Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Streamovací Server v prostředí Linuxu s podporou IPv6

Stream Server in Linux with IPv6 Support

2017 Jan Pejsar

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student: **Jan Pejsar**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R059 Mobilní technologie

Téma: Streamovací server v prostředí linuxu a s podporou IPv6
Stream Server in Linux with IPv6 Support

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem bakalářské práce je navrhnout řešení pro streamování v linuxovém prostředí s podporou IPv6. Server by měl být nízkonákladový, např. Raspberry Pi.

Řešení práce musí splňovat následující body:
1. Studium a popis streamování multimediálního obsahu.
2. Návrh řešení streamovacího zařízení postaveném na linuxu.
3. Ověření funkčnosti a měření síťových parametrů.
4. Zátěžové testování v laboratorních podmínkách.

Seznam doporučené odborné literatury:

Minoli, D. *Linear and Non-Linear Video and TV Applications: Using IPv6 and IPv6 Multicast* Wiley 2012, ISBN-13: 978-1118186589

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017

doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *28. dubna 2017*

.....
podpis studenta

Rád bych na tomto místě poděkoval všem, kteří mi pomohli s touto bakalářskou prací, své rodině a přítelkyni za morální a psychickou podporu, a hlavně vedoucímu mé bakalářské práce Ing. Pavlu Nevludovi, za jeho rady, připomínky a hlavně ochotu a trpělivost.

Abstrakt

Cílem této bakalářské práce je zaměřit se na návrh a implementaci nízkorozpočtového streamovacího serveru v prostředí Linuxu, který již bude podporovat Internet protokol verze 6, určený pro vysílání zábavního obsahu v reálném čase, jako jsou například seriály, filmy nebo živý streaming z web kamery.
K dosažení tohoto cíle bude použit jednodeskový počítač Raspberry Pi, který je jak rozměrově malý, tak cenově přijatelný. Jako Operační systém bude použit Raspbian, který bude nakonfigurován pro práci s adresací IPv6. Jako streamovací software bude použit program VLC, který může fungovat i jako klient i jako server. Na závěr práce vyhodnotím a shrnu naměřené parametry testování, které proběhnou ve školní laboratoři EB215.

Klíčová slova: Internet, IPv6, Linux, Raspbian, Raspberry, Sítě, Streaming, Video, VLC

Abstract

The main goal of the bachelor thesis is to focus on the design and implementation of cheap linux server running on OS Linux, which will support the Internet protocol version 6, dedicated to streaming of media in real time which consists of serials, movies, live streams from webcams.
In pursue of this I am using the Raspberry Pi, which is small and cheap one board system, the operating system used will be Raspbian configured for IPv6 usage. As a streaming software there will be VLC used as it can both be used as a server and a client. In conclusion I will summarize measured parameters and thesis itself, aswell as results obtained from testing which will conclude in schools lab EB215.

Keywords: Internet, IPv6, Linux, Networks, Raspbian, Raspberry, Streaming, Video, VLC

Seznam použitých zkratek a symbolů

AVI	–	Audio Video Interleave
CIDR	–	Classless Inter-Domain Routing
DNS	–	Domain Name System
DVR	–	Digital video recorder
FLV	–	Flash Video
GUI	–	Grafické uživatelské rozhraní
ICMPv6	–	Internet Control Message Protocol version 6
IETF	–	Internet Engineering Task Force
IPTV	–	Television over Internet protocol
IPsec	–	IP security
IPv4	–	Internet protocol version 4
IPv6	–	Internet protocol version 6
ISP	–	Internet service provider
NAT	–	Network Address Translation
NDP	–	Neighbor Discovery Protocol
MAC	–	Media Access Control
MKV	–	Matroska
MPEG TS	–	MPEG Transport Protocol
RPC	–	Remote procedure call
RFC	–	Request For Comments
RTMP	–	Real Time Messaging Protocol
RTSP	–	Real Time Streaming Protocol
SCTP	–	Stream Control Transmission Protocol
SLAAC	–	Stateless address autoconfiguration
TCP	–	Transmission Control Protocol
UDP	–	User Datagram Protocol
VLC	–	Video Lan Convertor
VLS	–	Video Lan Server
VOD	–	Video on demand

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Stejně jako v mnoha jiných oblastech informatiky,tak i v přenosu zábavního obsahu, jako jsou seriály, filmy či živé přenosy, dochází k revoluci přenosu přes protokol IP. Je stále více běžné, že lidé místo sledování klasické televize, zapnou stolní PC nebo laptop a sledují přenos buď živě, nebo se podívají na záznam již vysílaného programu. V současné době existuje mnoho streamovacích serverů, které umožňují sledovat jak živě, tak na dotaz. Mezi největší poskytovatele patří například Twitch.tv nebo Youtube.com.
V současné době se z již zastaralého a téměř vyčerpaného Internet protokolu verze 4 přechází na protokol verze 6. Tato práce má za úkol navrhnout vhodné řešení streamovacího serveru v prostředí Linuxu, které bude schopno vysílat zábavní obsah v reálném čase.
Na začátku této práce se budu zabývat základní problematikou streamování přes IP protokol, vysvětlím rozdíl mezi přenosem IPTV a klasickým televizním vysíláním. Následuje úvod do Internet protokolu verze 6, jaké jsou jeho hlavní výhody oproti verzi 4, vysvětlím co je multicasting, jaké jsou možnosti zabezpečení, rozšíření, soukromí atd. Dále se budu věnovat programu VLC, který je v bakalářské práci vybrán jako server i klient, rozeberu základní i pokročilé příkazy pro vysílání a příjem streamu. V práci popíšu dostupné formáty a možnosti streamování z tohoto programu.
Nakonec se budu zabývat vybraným hardwarem Raspberry Pi, který bude sloužit jako streamovací server a provedu měření ve školní laboratoři EB215, a to jak síťové za pomocí programu Slurm, tak i hardwarové za pomocí příkazu top. V závěru shrnu výsledky měření a z nich vyplyne verdikt, zdali je jak software VLC, tak hardware Raspberry Pi Model B+ 811-1284 vhodným řešením pro streamovací server.

2 Motivace

V současné době kdokoliv s širokopásmovým připojením k internetu může sledovat živé nebo skoro-živé televizní vysílání z 2250 různých kanálů ze zhruba 140 zemí a tato čísla narůstají každým měsícem. Nové technologie, zobrazovací paradigmata a distribuce obsahu přinášejí nové možnosti pro TV/video služby. Pět trendů, které jsou spojeny s novou generací doručování zábavního obsahu jsou pozorovatelné a mohou být aktivovány na straně poskytovatelů služeb, telekomunikačních společností, operátorů a IPSs. Tyto trendy jsou:

- Celosvětové zpřístupnění IPv6.
- Zprovoznění streamovacích a IPTV služeb.
- Migrace konzumentů ze sledování lineárních programů (v reálném čase) do sektoru nelineárního sledování (VOD - video on demand) programů).
- Větší spolehlivost a zájem o internetově produkovaný video obsah.
- Množství zařízení na kterých je možno video sledovat: TV obrazovka, personální PC monitor, tablet, herní konzole a chytré telefony.

V současné době mají uživatelé velké množství zařízení, ze kterých mohou zábavní obsah sledovat, ale zároveň se také mění sledovací návyky a požadavky diváků.
Z výzkumu společnosti Nielsen vyplynulo, že časový posun využití s používáním DVR (nelineární sledování nebo také Television On-Demand) se zvedá o 40% za několik předchozích let. Na Americkém trhu toto činí v průměru více než 8 hodin za měsíc oproti celkovému sledování TV, které činí 153 hodin za měsíc. Online sledování videa roste s tím jak si diváci vylepšují své PC, které podporují vyšší kvalitu rozlišení obrazu a také s tím, že širokopásmové připojení k internetu do domácnosti je mnohem stabilnější a rychlejší.
Ve Spojených státech Amerických v roce 2009 připadlo 29 hodin měsíčně Internetově založeného videa na diváka. Další studie od společnosti Nielsen také zjistila, že pouze sledování videa na mobilním telefonu roste průměrně rychlostí 50% ročně za předchozích pár let. Pomáhá tomu také fakt přechodu z broadcast na multicast a také vysílání obsahu konkrétní skupině uživatelů.
Očekává se, že v průběhu nadcházejících let, nastanou velké změny v infrastruktuře, která se používá pro doručení obsahu z TV na IP založené sítě přes kabel, satelit, 3G/4G bezdrátově, serverově založené distribuční systémy na poptávku. Hlavní sektory video distribuce se obávají, že pokud diváci budou stále více užívat DVR a Internetově založené video streamy, bude následovat eroze nebo přesun příjmů z reklam. Poskytovatelé infrastruktur si musí být pečlivě vědomi dopadu, které tyto vyvíjející se paradigmata budou mít na jejich sítě a jejich zdroj příjmů. Porozumění toho kam se technologie posouvá, pomůže posílit poskytovatelům svou pozici a využít tyto nové trendy ve svůj prospěch. [1]

3 Streamování

3.1 Definice Streamu

Základ slova streamování pochází z anglického slova stream, což znamená proud. Můžeme jej definovat jako technologii přenosu audiovizuálního materiálu mezi zdrojem (např. server) a koncovým uživatelem (např. běžné PC). V současnosti se streamování používá zejména pro přenos zvuku nebo videa přes internet (webcasting).

3.2 Kvalita

U streamování řešíme také kvalitu daného toku dat, která je dána použitým formátem audia nebo videa.

3.2.1 Zvuk

U audia se nejčastěji využívá formátů MP3, OGG, AAC+ a WMA obvykle v datových tocích 16-256 Kbps. Audio můžeme streamovat jako single bitrate, což je vlastně jeden jediný konstantní datový tok a nebo multibitrate, což je více konstantních datových toků, které přenášíme dohromady v jednom datovém proudu mezi kodérem a serverem. Klientské zařízení, který přehrává multibitrate stream, dokáže potom měnit kvalitu zvuku v případě zlepšení nebo zhoršení kvality připojení k internetu posluchače automaticky.

3.2.2 Video

Pro přenos videa po internetu je potřeba kodeků pro zmenšení objemu dat. Pro streamování se využívá nejvíce flashových kodeků, Windows Media, Real Time, Quick time a MPEG-4. Přenos záznamu v rozlišení 720x576 býval velmi náročný, z toho důvodu se nejvíce používalo streamování v rozlišení 320x240 s datovým tokem 256-512 Kbps. V současné době je tomu jinak a streamuje se v rozlišení full HD nebo 4K, které nabízí například služba Youtube nebo Netflix.

3.3 Webcasting

Webcasting může probíhat buď v reálném čase (např. rádio, internetová televize), nebo způsobem Video on demand (např. YouTube). Pokud chceme streamovat video nebo audio pro více uživatelů, potřebujeme kromě vlastního obsahu také streamovací server, který se postará o komunikaci s cílovým počítačem nebo počítači a zajišťuje plynulý přenos dat. Webcasting se hodně používá například v komerčním sektoru pro přenos prezentací, e-learningu a dalších komunikačních aktivit. [10]

3.4 RTMP

RTMP (Real-Time Messaging Protocol) byl původně vyvinut společností Macromedia pro streamování audia, videa a dat přes Internet mezi Flash přehrávačem a serverem. Macromedia je nyní ve vlastnictví společnosti Adobe, která protokol zpřístupnila pro veřejné použití. Protokol RTMP má několik variant a těmi jsou:

- Čistá varianta protokolu, která pracuje na povrchu a používá defaultně TCP port s číslem 1935.
- RTMPS, je RTMP přes TLS/SSL připojení.
- RTMPE, je RTMP kryptované za použití zabezpečovacího mechanismu od Adobe.
- RTMPT, je zapouzdřené do HTTP žádostí, aby prošel firewallem, RTMPT používá cleartext žádosti na TCP portech 80 a 443, aby se vyhnul nejčastěji filtrování provozu. Zapouzdřené pakety pak mohou nést buď čisté pakety RTMP, RTMPS nebo RTMPE.
- RTMFP, je RTMP přes UDP namísto TCP, tento protokol umožňuje komunikaci koncových uživatelů P2P (peer to peer).

RTMP je protokol na základě TCP, který udržuje konstantní spojení a dovoluje komunikaci s nízkým zpožděním. Rozděluje stream do fragmentů z důvodu přenosu co nejvíce informací pokud možno co nejplynulejším způsobem, velikost fragmentů je dohodnuta dynamicky mezi klientem a serverem. Většinou se velikost nemění, standardní velikost je 64 bytů pro audio data a 128 bytů pro video data a spoustu dalších datových typů. Data z různých streamů pak mohou být prokládána a multiplexována přes jedno spojení. S delšími datovými segmenty protokol tudíž nese jenom jednobytovou hlavičku pro každý fragment. Požaduje tedy velmi malou režii. V praxi individuální fragmenty prokládány nejsou. Prokládání a multiplexování se dělá až na paketové úrovni, s několika RTMP pakety napříč několika aktivními kanály prokládanými tak, aby se zajistilo, že je použita celá šíře pásma daného kanálu, je zde co nejmenší zpoždění a další specifikace kvality služby. Takto prokládané pakety jsou považovány za nerozdělitelné a nejsou prokládány na fragmentační úrovni.
RTMP definuje několik virtuálních kanálů, přes které je možno posílat a přijímat pakety, a které operují nezávisle na sobě. Pro příklad, je zde kanál, který řeší RPC žádosti a odpovědi, kanál pro data video streamů, kanál pro audio data, kanál pro kontrolní odpovědi

mimo pásmo atd. Během typického RTMP spojení, může být několik kanálů aktivních zároveň v jakémkoli daném čase. Když jsou RTMP data zakódována, vygeneruje se hlavička paketu. Hlavička mimo jiné také specifikuje ID kanálu, přes který se má daný paket poslat, časovou známku kdy byl paket vygenerován a zátěž paketu. Po této hlavičce následuje skutečný obsah paketu, který je poté fragmentován podle dohodnuté velikosti než je odeslán do sítě. Hlavička se nefragmentuje a její velikost se nepočítá mezi data v prvním fragmentu paketu. Jenom aktuální obsah paketu se počítá do fragmentace. Na vyšších úrovních protokol RTMP zapouzdřuje MP3 nebo ACC audio a FLV1 multimediální streamy a může dělat RPC (Remote procedure calls) za použití formátu aktivních zpráv. Jakékoliv potřebné RPC služby jsou řešeny asynchronně, za použití jednoho klient/server požadavek/odpověď modelu, komunikace v reálném čase není nutná.

3.5 RTSP

RTSP (Real Time Streaming Protocol) je protokol na aplikační vrstvě pro kontrolu nad přenosem dat v reálném čase. RTSP poskytuje rozšiřitelný framework aby umožnil kontrolované, na požadavek, doručení dat v reálném čase, jako je například audio nebo video. Zdroje dat mohou obsahovat jak reálné datové zdroje tak uložené datové záznamy. Tento protokol je určen pro kontrolu nad několika datovými přenosy a poskytuje možnost výběru přenosového kanálu jako jsou například UDP, multicast UDP a TCP, a poskytuje způsoby výběru doručovacího mechanismu na základě RTP. RTSP zahajuje a kontroluje buď jeden nebo více časově synchronizovaných streamů medií jako jsou audio nebo video. Nedoručuje tento konstantní stream sám o sobě, ale je zde možnost prokládání dat, RTSP se chová jako síťový ovladač pro multimediální servery. Neexistuje žádný pojem jako RTSP spojení, místo toho, server udržuje relaci označenou identifikátorem. RTSP relace není v žádném případě spojená se spojením na transportní úrovni jako je například TCP spojení. Během této relace, RTSP klient může otevřít nebo uzavřít mnoho spolehlivých transportních spojení k serveru pro zadání RTSP žádostí, alternativně může také použít také protokol UDP. [12]

3.6 MPEG TS

MPEG-TS (Transport Stream) je jeden ze standardních kontejnerových formátů pro přenos a ukládání audia a videa, je používán pro vysílací systémy jako jsou například IPTV. Transportní stream specifikuje kontejnerový formát a zapouzdřuje pakety do elementárních streamů s opravou chyb a synchronizačními vlastnostmi, aby byla zachována integrita přenosu pokud je signál degradován.
Transportní stream v podstatě zapouzdří počet jiných substreamů, často to jsou elementární paketové streamy a spojí hlavní datový tok MPEG kodeku a dalších několik datových toků, jako například DTS nebo AC3 audio, či dokonce titulky k přenášenému filmu. Můžeme ovšem vysílat také tabulky, které identifikují dané streamy a nebo jiné specifické informace, jako je například manuál k TV.

V praxi to funguje tak, že několik televizních kanálů je sdruženo do jednoho streamu. Každý stream je rozdělen do sekcí po 188 bytech a jsou prokládány mezi sebou Díky malé velikosti paketu mohou být streamy prokládány s menším zpožděním a lepší odolností vůči chybám na rozdíl od AVI nebo MKV, které zabalí rámec do jednoho paketu. Toho se využívá například u videokonferencí, kde jediný velký rámec může způsobit nepřijatelné zdržení zvuku. [14]

3.6.1 Formát paketu

Každý paket začíná synchronizačním bytem a hlavičkou, která může být následována dalšími volitelnými hlavičkami, zbytek paketu obsahuje konkrétní data. Všechny hlavičkové soubory jsou čteny jako Big Endian. Pakety mají délku 188 bytů, nicméně komunikační médium může přidat další informace.

3.7 SCTP

SCTP (Stream Control Transmission Protocol) je protokol transportní vrstvy, který plní podobnou roli jako protokoly TCP a UDP, je standardizován skupinou IETF v RFC4960. Poskytuje některé služby, které jsou již definované jak v UDP tak TCP, je zprávami orientovaný jako UDP a zajišťuje spolehlivý, sekvenční přenos s řízením přetížení jako TCP. Od těchto dvou se liší tím že poskytuje tzv. multihoming a redundantní cesty, které zvyšují spolehlivost a odolnost.
Aplikace používající SCTP poskytnou svoje data k přenosu ve zprávách (skupinách bytů) na SCTP transportní vrstvu. Protokol umístí zprávy a kontrolní informace do separátních datových a kontrolních částí, které jsou pak identifikovány částí hlavičky. Protokol může fragmentovat zprávu do více datových částí, ale každá tato část obsahuje data pouze od jedné uživatelské zprávy. SCTP pakety se pak skládají z SCTP částí. SCTP paket poté obsahuje hlavičku, kontrolní části a poté následují části datové.
SCTP umožňuje multi-streamování, což znamená schopnost přenášet paralelně několik nezávislých streamů dat. Na rozdíl od TCP, které zachovává tok bytů v streamu tím, že zahrne sekvenční číslo do každého segmentu, SCTP na druhou stranu přiřadí sekvenční číslo ke každé zprávě poslané přes stream, a to umožňuje nezávislé seřazení zpráv v různých streamech. [19]

3.7.1 Formát paketu

SCTP pakety mají jednoduší základní strukturu než TCP pakety, skládají se ze dvou základních složek:

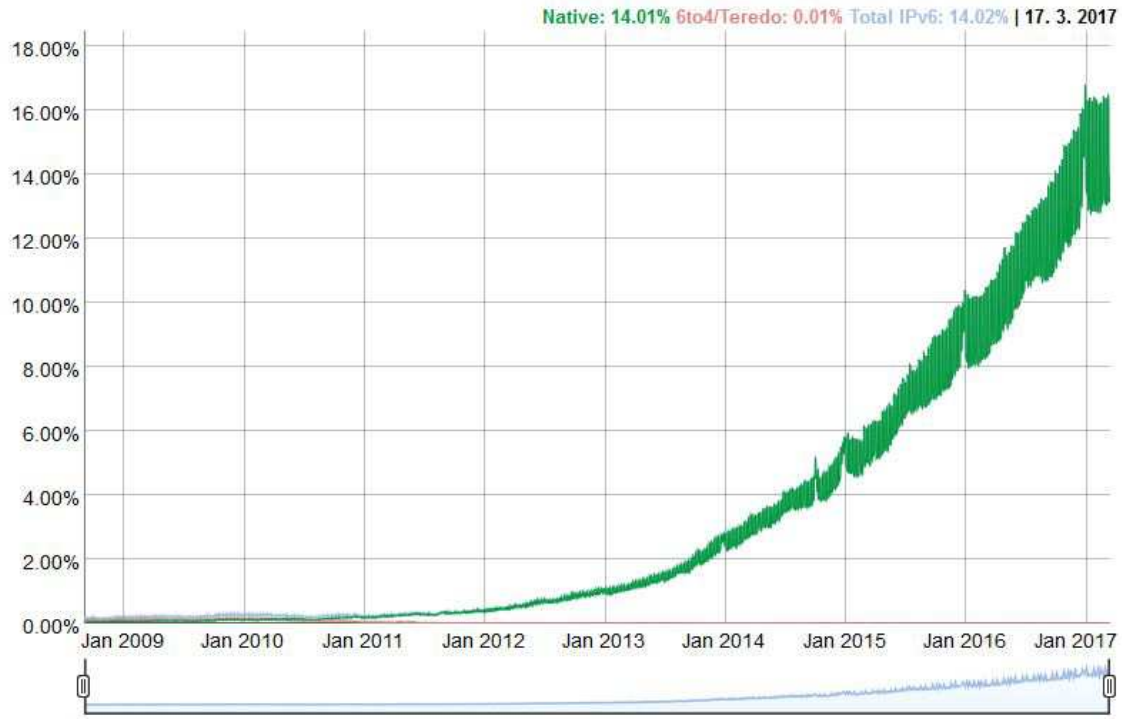
- Běžná hlavička, která má velikost prvních 12 bytů.
- Datové části, které zahrnují zbývající část paketu.

Každý datový kus začíná jedním bytovým identifikátorem, celkem je 15 typů datových částí definovaných v RFC4960. 8 flag bitů, 2 bytové pole délky a data se skládají ze zbytku části, pokud celková část nemá více než 4 byty, je doplněna nulami, které nejsou zahrnuty v délce části. 2 bytové pole délky limituje každou část na maximálně velikost 65535 bytů.

4 IPv6

4.1 Internet Protokol verze 6

IPv6 je nejnovější dostupná verze internetového protokolu IP, komunikačního protokolu, který poskytuje identifikaci a lokalizaci systémů pro počítač v sítích a směruje provoz napříč internetem. IPv6 byla vyvinuta společností Internet Engineering Task Force (IETF) jako řešení dlouho předvídanému problému vyčerpání adresního prostoru IPv4.
IPv6 má v budoucnu nahradit IPv4. Každé zařízení, které se nachází na internetu má přiřazenou unikátní IP adresu pro identifikaci a lokační definici. S rychlým růstem používání internetu po zpřístupnění veřejnosti v roce 1990, začalo být zřejmé, že bude potřeba mnohem více adres pro připojování zařízení, než dovoloval adresní prostor IPv4. IETF představila tento nový protokol v roce 1998.
IPv6 používá 128 bitovou adresu, teoreticky dovoluje 2^{128} nebo přibližně $3,4 * 10^{38}$ adres. Konkrétní číslo je o něco menší, protože několik rozsahů je rezervováno pro speciální užití nebo úplně vyřazeno z běžného používání. Celkový počet IPv6 adres je $7,9 * 10^{28}$krát větší než počet adres IPv4, které používají 32 bitové adresy a poskytuje zhruba 4,3 miliard adres.
Oba protokoly nejsou konstruovány, aby byly interoperabilní, a to komplikuje přechod na IPv6. Nicméně několik IPv6 převodových mechanizmů bylo vyvinuto, aby byla umožněna komunikace mezi IPv4 a IPv6 hosty. [3]



Obrázek 4.1: Rozšíření IPv6 celosvětově podle statistik uživatelů Google k datu 17.3 2017. [11]

4.2 Porovnání oproti IPv4

Krom zvětšeného adresního prostoru poskytuje IPv6 další technické benefity. Konkrétněji, dovoluje metody hierarchického přidělování adres, které usnadňují agregaci trasy napříč internetem, tím pádem limituje expanzi směrovacích tabulek. Použití multicast adresace je rozšířeno a zjednodušeno a umožňuje další optimalizace pro doručovací služby. Mobilita zařízení, bezpečnost a konfigurační aspekty byly zváženy při tvorbě protokolu taktéž. IPv4 a IPv6 můžeme porovnat například:

- Větší adresní prostor.
- Multicasting.
- SLAAC.
- IPsec.
- Mobilita.
- Možnosti rozšiřitelnost.
- Soukromí.

4.2.1 Větší adresní prostor

Hlavní výhodou IPv6 oproti IPv4 je její větší adresní prostor. Délka IPv6 adresy je 128 bitů v porovnání s 32 bity, které má adresa IPv4, adresní prostor IPv6 je tudíž teoreticky 2^{128} to je $3,4 * 10^{38}$ adres. Navíc adresní prostor IPv4 je nevhodně alokovaný, v roce 2011 je užíváno pouze 14% dostupných adres. Ačkoliv jsou tato čísla velká, nebylo úmyslem vývojářů IPv6 adresního prostoru zajistit geografickou saturaci s použitelnými adresami. Delší adresy spíše usnadňují alokaci adres, umožňují výhodnou agregaci cest a dovolují implementaci speciálních funkcí adres.
V IPv4, komplexní Classless Inter-Domain Routing (CIDR), byly vyvinuty metody, aby se co nejlépe využil malý adresní prostor. Standardní velikost podsítě v IPv6 je 2^{64} adres, to poskytuje daleko větší rozsah, než má celý adresní prostor IPv4. Skutečné užití adresního prostoru IPv6 bude procentuálně malé, ale síťová správa a směrování jsou vylepšeny velkým adresním prostorem a hierarchickou směrovací agregací.
Hlavní snahou IPv4 je přečíslování existující sítě pro nového síťového poskytovatele s jinými směrovacími prefixy. U IPv6 změna prefixu oznámena několika routery může z principu přečíslovat celou síť, protože identifikátor hosta, nejméně významných 64 bitů adresy, může být nezávisle konfigurován hostem.

4.2.2 Multicasting

Multicasting je přenos paketů z jednoho bodu do více míst jedinou odesílací operací. Multicasting je částí základní specifikace IPv6. V IPv4 je toto volitelná, ale běžně implementovaná vlastnost. IPv6 multicast adresace sdílí základní vlastnosti a protokoly s IPv4 multicastem a navíc přináší změny a vylepšení tak, že odstraňuje potřebu některých protokolů.
IPv6 neimplementuje tradiční IP broadcast a ani jej nedefinuje. V IPv6 stejného výsledku může být dosaženo odesláním paketů na link-local all nodes multicast group na adrese ff02::1, což je podobné IPv4 multicastingu na adrese 224.0.0.1. IPv6 také poskytuje nové multicast implementace, zahrnující vkládání adres místa setkání v IPv6 multicast skupinových adresách, což zjednodušuje rozestavění interně-doménových řešení.
V IPv4 je dosti složité, aby organizace získala jednu globální směrovatelnou skupinu a implementace interně-doménových řešení je skoro nemožná. Unicast adresy přiřazené lokálními internetovými registry pro IPv6 mají nejméně 64 bitový směrovací prefix, z toho vyplývá nejmenší možná velikost sítě v IPv6 64 bitů. S tímto přiřazením je možné vložit prefix adresy unicast do IPv6 multicast adresní formy a zároveň poskytnout 32 bitový blok (nejméně významné bity adresy) nebo 4,2 miliard multicastových identifikátorů skupin. Tudíž má každý uživatel IPv6 podsítě automaticky dostupnou skupinu globálně směrovatelných zdrojově specifických multicastových skupin pro multicast aplikace.

4.2.3 SLAAC

SLAAC (Stateless Address Autoconfiguration) je v překladu bezstavová adresní autokonfigurace. IPv6 hosté se mohou konfigurovat automaticky po připojení k IPv6 síti za použití NDP (Neighbor Discovery Protocol) skrze ICMPv6 (Internet Control Message Protocol version 6) zprávami objevu směrovače. Po prvním připojení do sítě host pošle linkově-lokální zprávu zvanou router solicitation multicast request pro jeho konfigurační parametry. Routery odpoví na takovouto žádost paketem router advertisement, který obsahuje konfigurační parametry internetové vrstvy.
Pokud je bezstavová adresní autokonfigurace nevhodná pro aplikaci, síť může použít stavovou konfiguraci s DHCPv6 (Dynamic Host Configuration Protocol version 6) a nebo může být host nakonfigurován manuálně za použití statických metod. Směrovače umožňují speciální případ požadavků pro adresní konfiguraci, protože jsou většinou zdroje informací pro autokonfiguraci, jako jsou router a prefix advertisements. Bezstavové konfigurace routerů může být dosaženo speciálním protokolem pro přečíslování.

4.2.4 IPsec

IPsec (Internet Protocol Security) byl původně vyvinuto pro IPv6, ale prvotně se výrazně používal v IPv4, pro který byl upraven. IPsec byl povinnou součástí IPv6 protokolu, ale nyní je pouze volitelný.

4.2.5 Mobilita

Na rozdíl od mobilní IPv4, se IPv6 vyhýbá trojúhelníkovému směrování. IPv6 směrovače umožňují celým podsítím přesunout se na nový směrovací bod spojení bez zbytečného přečíslování.

4.2.6 Možnosti rozšiřitelnosti

Hlavička IPv6 paketu má minimální velikost 40 oktetů. Možnosti jsou implementovány jako rozšíření, což umožňuje v budoucnosti rozšířit protokol bez zásahu do základní paketové struktury.

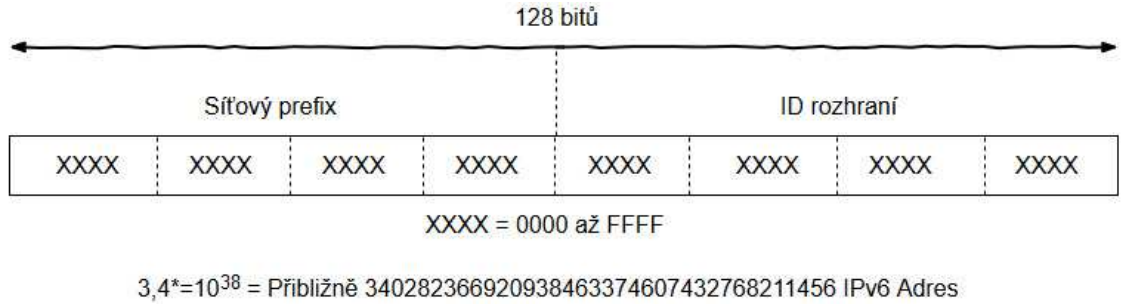
4.2.7 Soukromí

Stejně jako IPv4, IPv6 podporuje globální unikátní IP adresy, podle kterých může být aktivita každého zařízení dohledána v síti. Design IPv6 měl znovu zdůraznit end-to-end princip síťového návrhu, který byl původně koncipován během zřízení časného Internetu. V tomhle náhledu má každé zařízení v síti svoji unikátní globální adresu dostupnou z jakékoli jiné lokace na Internetu. Sledování síťového prefixu je menší problém, pokud

ISP uživatele přiřadí uživateli dynamický síťový prefix přes DHCP. Rozšíření soukromí vykoná málo pro ochranu uživatele před nalezením, pokud ISP přidělí statický síťový prefix. V tomto případě je síťový prefix unikátní identifikátor pro sledování a identifikátor rozhraní je sekundární. V IPv4 snaha o zachování adresního prostoru s použitím NAT (Network Address Translation) skrývá síťový adresní prostor, hosty a topologie. Pokud užíváme u IPv6 adresní auto-konfiguraci, identifikátor rozhraní (MAC adresa) portu rozhraní je použit, aby zajistil, že jeho veřejná IP adresa bude unikátní, odhaluje typ používaného hardwaru, a umožňuje tak unikátní řízení pro uživatelskou aktivitu online.
Není ovšem požadavkem aby IPv6 hosté používali adresní auto-konfiguraci, i když není adresa vytvořena na základě MAC adresy, adresa rozhraní je globálně unikátní, v kontrastu k NAT maškarádě privátních sítí. Rozšíření soukromí bylo v IPv6 vytvořeno, aby tyto problémy adresovalo. Pokud je rozšíření soukromí povoleno, operační systém generuje náhodně identifikátory hosta pro zkombinování s přiřazeným prefixem sítě. Tyto efemérní adresy jsou použity pro komunikaci se vzdálenými hosty, stopování jednoho zařízení je posléze složitější.
Některé distribuce Linuxu, Windows (XP a výše) OS X (10.7 a výše) mají rozšíření soukromí aktivováno standardně. Jako bonus k dočasným adresám, rozhraní obdrží také stabilní adresu. Identifikátory rozhraní jsou generovány tak, aby byly stabilní pro každou podsíť, ale aby se měnily, když host přechází z jedné sítě do jiné. Tímto způsobem je těžké sledovat hosta, který se přesouvá ze sítě do sítě. V jedné konkrétní síti bude mít host vždy stejnou adresu, aby síťový přístup a kontrola mohly být potencionálně nakonfigurovány. Tradiční metoda generování identifikátorů rozhraní k použití pro unikátní adresní přiřazení byla založena na MAC adresaci.
Vzhledem k efektivnějším ochranám soukromí nebyla tato metoda schválena v některých operačních systémech a byla nahrazena nově vytvořenými metodami RFC7217. Rozšíření soukromí nechrání uživatele před jinými formami sledování na jiných vrstvách jako jsou například sledování cookies na aplikační vrstvě.[5]

4.3 Tvar IPv6

IPv6 adresa je reprezentována jako osm skupin, každá skupina obsahuje čtyři hexadecimální číslice.
Skupiny jsou odděleny dvojtečkami pro příklad 2001:db8:0000:0000:0000:0032:8a2e:ffff, existují konvence pro zápis IPv6, například zápis adresy v kanonické podobě by byl 2001:db8::32:8a2e:ffff. [4]



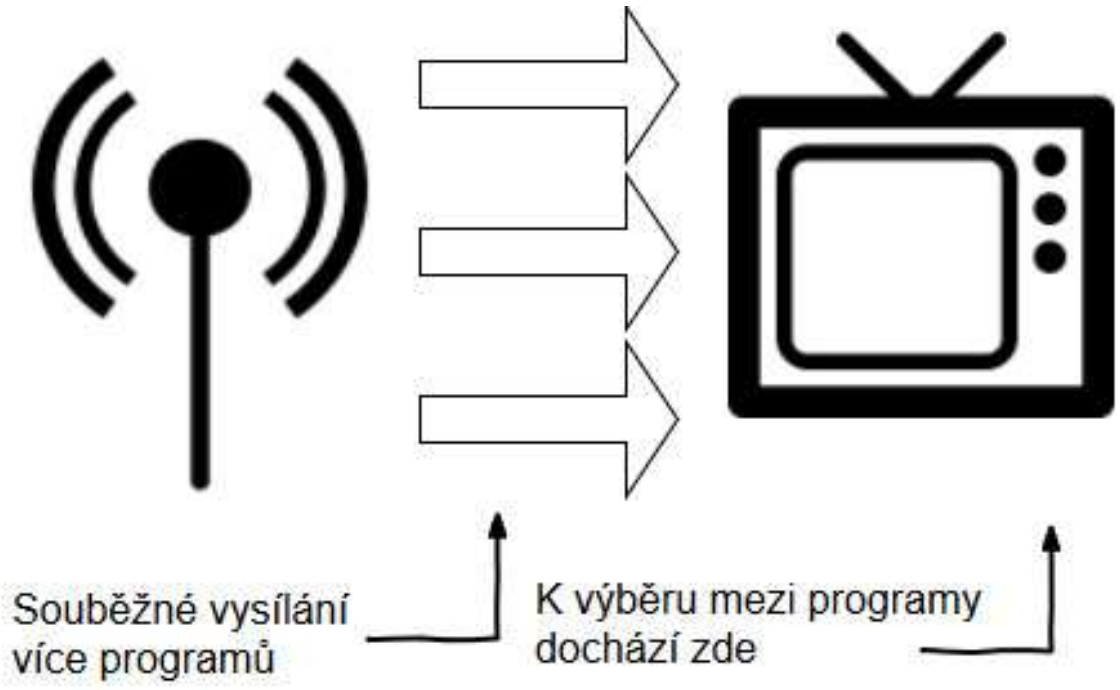
Obrázek 4.2: Tvar IPv6 - Síťový prefix a ID rozhraní.

5 IPTV

IPTV je zkratka pro Internet protocol television. Přenos televizního obsahu je tedy řešen pomocí protokolu IP, narozdíl od tradičního pozemního, satelitního nebo kabelového vysílání. Pro srovnání si nejdříve uděláme shrnutí tradičního televizního vysílání. [5]

5.1 Tradiční vysílání

V angličtině jej označujeme jako Broadcasting, což znamená souběžné vysílání ke všem potencionálním příjemcům. V podstatě to také znamená, že jde o souběžné vysílání více programů. Výběr mezi programy je proveden až na straně příjemce na základě jeho preferencí.



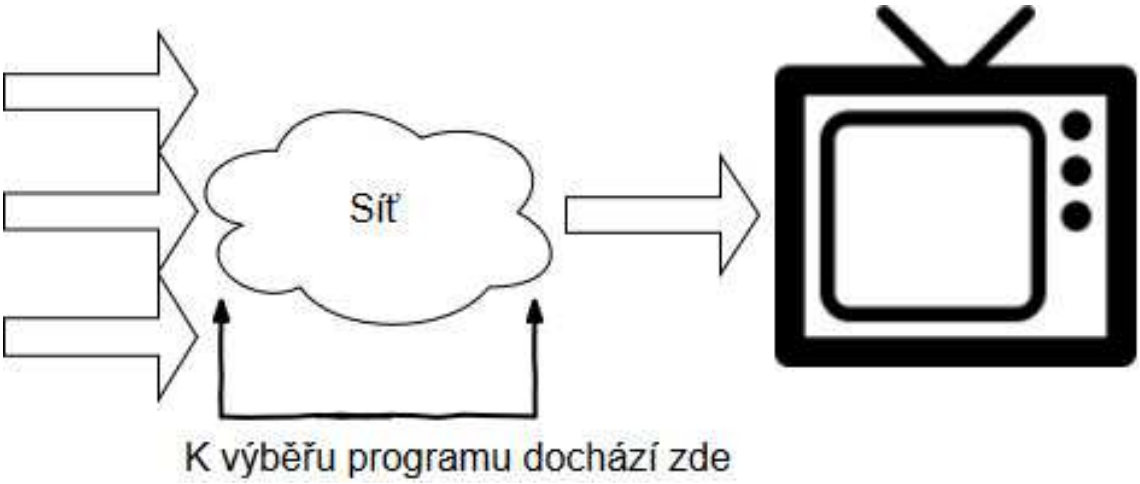
Obrázek 5.1: Představa klasického televizního vysílání.

Nutno podotknout, že broadcasting vysílání je pouze jednosměrné a nelze od vysílatele zjistit, kolik má aktuálně diváků a kdo se dívá na jaký program. Pro průzkumy sledovanosti se tedy v tomto případě musí používat peoplemetry nebo zpětné dotazy. Technicky se zde používá přenosová cesta, která vede ke koncovým uživatelům a je dimenzovaná tak, aby zvládla souběžný přenos více programů.
Nicméně to v praxi neplatí, a proto přes kabel, satelit a éter se dá šířit jen omezený počet programů. Z toho vznikají konflikty o místa na transpondéru či kabelu, boj o licence a

hlavně snaha od digitalizaci vysílání, které využívá efektivněji kapacitu pásma a zvládne přenést více programů souběžně. Další charakteristická vlastnost klasického způsobu vysílání je jeho stejnost pro všechny příjemce. Jedná se pouze o jednosměrnou distribuci jediného a toho samého signálu ke všem příjemcům, všichni tudíž dostávají totožný obsah. Ten není možno nijak diferencovat, přizpůsobit individuálně jednotlivým příjemcům podle jejich preferencí. Co lze nanejvýše udělat je, že do toku dat přijímaného všemi, vložíme nějakou informaci, která je zakódována nebo zabezpečena jiným způsobem a dekódovat ji může pouze konkrétní osoba nebo skupina lidí.
Dalším důsledkem tohoto vysílání je například, že z kapacitních důvodů a také kvůli absenci zpětné vazby, není prostor pro nelineární média. Služba VOD (Video on Demand) je pro programy a služby, které nemají stanovené vysílací schéma určené vysílatelem, ale které si vyberou a určí jednotliví příjemci podle toho, kdy mají čas a co preferují. Další důsledek je například ten, že reklamy nemohou být přizpůsobeny pro konkrétní osobu nebo skupinu diváků.

5.2 Princip IPTV

V současné době má IPTV od klasického vysílání dvě základní odlišnosti. První odlišností je, že vysílání již není jednosměrné, je zde existence plnohodnotné zpětné vazby mezi příjemcem a vysílatelem, což otevírá dveře interaktivitě a nelineárním službám. Umožňuje to také vysílateli přesně identifikovat diváka a komunikovat s ním, díky čemuž může určit jeho preference a následně může mít jistotu individualizace a personalizace, například zařazením specifické reklamy.



Obrázek 5.2: Představa funkčnosti přenosu IPTV.

Druhá zásadní odlišnost od klasického vysílání spočívá v tom, že přenosový kanál v minulosti nebyl dostatečně dimenzován, aby k divákovi přenášel více programů současně. Je zde většinou umožněn přenos pouze jednoho jediného programu. Důsledkem pak je,

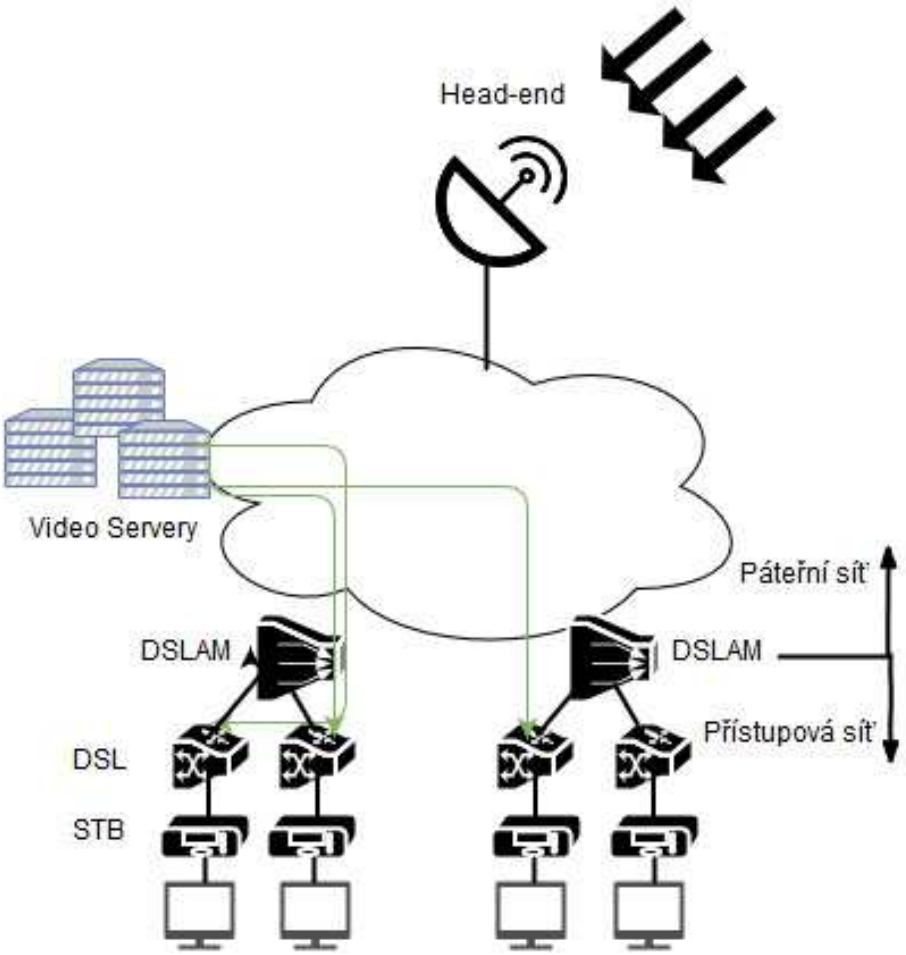
že i když má uživatel doma více zařízení pro sledování programů a ty běží současně, mohou všechny přehrávat jen jeden program současně.
Signál lze ovšem rozbočit a rozvést k více zobrazovacím jednotkám, ale zdroj signálu je pouze jeden a nabízí jeden jediný program. To ovšem neznamená že se jedná o omezení IPTV jako takové, které by se nikdy nedalo měnit. Naopak v současné době je již větší kapacita přenosového kanálu a jiné technologie. Toto omezení se týkalo pouze těch variant IPTV, které byly provozovány na přípojkách ADSLm kde se vešel jeden jediný program. Například ADSL2+, které má větší přenosové kapacity, umožňuje k divákovi přenášet více programů souběžně. Větší kapacity lze dosáhnout například ještě optickými přípojkami.

5.3 Multicast u IPTV

Je třeba rozvětvit jeden přicházející stream do určitého počtu streamů, a to se v počítačových sítích řeší prostřednictvím multicastu.
Multicast se používá také v prostředí IPTV. Každý jednotlivý program představuje jednu multicastovou skupinu a příjemci jsou členy v jedné z těchto multicastových skupin, podle toho jaký obsah si přejí sledovat a ten je pak vysílán na jejich zařízení. Přepnutí mezi programy je potom řešeno přechodem z jedné multicastové skupiny do skupiny jiné. Páteřní sítě jsou proto vybaveny dostatečnou inteligencí i na svých okrajích (DSLAM), aby potřebný multicasting zvládly.
Stejně tak, když chce provozovatel vkládat individuální reklamy, musí to provést také zde. Takto jsou řešeny u IPTV služby živého vysílání. Na stejném principu jsou řešeny i služby PPV (Pay Per View), to je možnost sledovat jednotlivé pořady na placené bázi, tedy například hokejový přenos s daným vysílacím schématem.

5.4 Nelineární služby

Jinak tomu je se službami, které mají nelineární charakter, a u kterých si od začátku o délce obsahu rozhoduje uživatel sám. Výsledkem je tedy individuální datový stream, který je nutno přenést od zdroje až ke koncovému příjemci.
Nemá zde vůbec žádný smysl sdílení datových toků a jejich následné rozvětvování. Toto se týká služeb charakteru Video on Demand, například IPTV od TO2CR, která má svoji službu videotéka, kde si uživatel může přehrávat určité konkrétní videa v jím zvoleném čase. Jedná se také o videa ze záznamu, která jsou také realizována na straně poskytovatele.
U všech nelineárních služeb je poskytovatel motivován k tomu, aby jeho vybavení, především servery generující video streamy, umístil co nejblíže k přístupové síti a vytvořil tím co možná nejmenší zátěž na své páteřní sítě.



Obrázek 5.3: Představa šíření nelineárního obsahu v sítích IPTV.

6 VLC

VLC media player, běžně známý pod názvem VLC, je přenosný, zdarma a open-source, multiplatformní přehrávač a streamovací server, napsán VideoLan projektem. VLC je dostupné jak pro desktopové operační systémy tak pro mobilní zařízení jako jsou například Android, Windows mobile, iOS, atd. Software VideoLAN vznikl jako akademický projekt v roce 1996. Zkratka znamená VideoLan Client, když VLC byl ještě jenom klient VideoLAN projektu.
V současné době VLC již není klient, proto tento název již neplatí. Původně bylo zamýšleno, že bude používán jako klient a server pro streamování videa ze satelitů napříč sítí na kampusu. Původně vyvinut studenty École Centrale Paris, nyní je vyvíjen lidmi z celého světa a koordinován neziskovou organizací VideoLAN. V roce 1998 byl od základů přepsán a vydán pod GNU (General Public License) 1. Února roku 2001. Funkcionalita VideoLAN serveru (VLS) byla přesunuta do programu VLC a VLS se již nepoužívá. Projekt se také přejmenoval na VLC media player, protože klient a server jsou nyní součástí jednoho programu. [2]

6.1 Formáty

VLC podporuje mnoho audio a video kompresních metod a souborových formátů. Je také schopen streamovat media přes počítačové sítě a transkódovat multimediální soubory. Základní distribuce obsahuje poměrně hodně kódovacích a dekódovacích knihoven, které jsou zdarma. Vyhýbá se tedy potřebě hledání nezbytných pluginů. Knihovna libavcodec, vytvořená FFmpeg projektem poskytuje hodně VLC kodeků, ale primárně program používá své vlastní muxery a demuxery. Má také svoje vlastní protokolové implementace. [6]

Tabulka 6.1: Audio a Video formáty podporované programem VLC.

Audio Formáty	MPEG Layer 1/2, MP3 - MPEG Layer 3, AAC - MPEG-4 part3, Vorbis, AC3 - A/52, E-AC-3, MLP / TrueHD¿3, DTS, WMA 1/2, WMA 3, FLAC, ALAC, Speex, Musepack / MPC, ATRAC 3, Wavpack, Mod, TrueAudio, APE, Real Audio, Alaw/ulaw, AMR (3GPP), MIDI, LPCM, ADPCM, QCELP, DV Audio, QDM2/QDMC, MAC
Video Formáty	MPEG-1/2, DivX® (1/2/3/4/5/6), MPEG-4 ASP, XviD, 3ivX D4, H.261, H.263 / H.263i, H.264 / MPEG-4 AVC, Cinepak, Theora, Dirac / VC-2, MJPEG (A/B), WMV 1/2, WMV 3 / WMV-9 / VC-1, Sorenson 1/3, DV, On2 VP3/VP5/VP6, Indeo Video v3 (IV32), Real Video (1/2/3/4).

6.2 Streamování pomocí VLC

Existuje několik možností jak vysílat nebo přijímat stream. Můžeme jej nastavit v jednoduchém GUI, a nebo jej můžeme spustit z terminálu či příkazové řádky, zadáním od jednoduchého až po několikařádkového příkazu. [7]

6.2.1 Streaming Wizard GUI

Pro Streaming Wizard akceptuje IPv6 adresy mezi hranatými závorkami, např. [2001:db8:ac:1242:211:11ff:fe25:e6b4]. Pokud sepcifikujeme linkovou lokální adresu, musíme také specifikovat síťové rozhraní, které používáme např. [2001:db8:ac:1242:211:11ff:fe25:e6b4%eth0] aby se připojilo rozhraní eth0. Pokud streamujeme přes protokol HTTP, IPv6 je automaticky použita jako defaultní, takže oba klienti jak IPv4 tak IPv6 jsou povoleny. Pokud chceme specifikovat DNS jméno hosta, VLC upřednostňuje IPv4 řešení. Musíme tedy specifikovat jméno hosta, které řeší jen IPv6 adresy, nebo povolit možnost Force IPv6 v pokročilém nastavení v Preferences/General Settings/Input. [**?**]

6.2.2 Z příkazové řádky

–ipv6 možnost v terminálu, zajišťuje prioritní použití IPv6 před IPv4.

```
vlc −vvv video1.xyz −−ipv6 −−sout udp:[ff08::1] −−ttl 12
```

kde následující parametry znamenají:

- video1.xyz je soubor, který chceme streamovat, můžeme také vložit dvdsimple:/dev/dvd pro streamování DVD nebo jiné vstupní jednotky.
- ff08::1 je buď
  - IPv6 adresa PC na který chceme vysílat Unicast.
  - IPv6 Multicast adresa.
- 12 je hodnota Time to Live IP paketů, což znamená že náš stream projde 11 routery než jsou pakety zahozeny.

Můžeme také specifikovat výstupní rozhraní, kde eth0 je název rozhraní.

```
vlc −vvv video1.xyz −−ipv6 −−sout udp:[ff08::1%eth0] −−ttl 12
```

6.3 Příjem Streamu

Stream můžeme přijmout několika různými způsoby, mezi které například patří.

6.3.1 Streaming Wizard GUI

Výběr streamu v File/Open Network Stream. Pro přijetí UDP/RTP unicast streamu odeslaného do konkrétního PC, bychom měli vybrat možnost Force IPv6 a také případně vybrat zdrojový UDP port. Pro příjem UDP multicast streamu vybereme UDP/RTP multicast možnost a specifikujeme multicast adresu uvnitř hranatých závorek. Syntaxe IPv6 adresy je stejná jako při odesílání streamu.

6.3.2 Z příkazové řádky

Stejně jako při odesílání, –ipv6 nám umožňuje použít IPv6 defaultně před IPv4.

```
vlc −vvv −−ipv6 udp:@[ff08::1]
```

Stejně jako při odesílání je možno specifikovat výstupní rozhraní.

```
vlc −vvv −−ipv6 udp:@[ff08::1%eth0]
```

kde eth0 je název výstupního rozhraní.

6.4 Stream přes stream output

Stream output je název vlastnosti ve VLC, která umožňuje výstup jakéhokoli streamu do souboru nebo jako síťový stream. Během tohoto procesu mohou být použity různé způsoby zpracování pro tok dat. Patří mezi ně například transkódování, změna velikosti, filtry, re-muxing. Stream output obsahuje různé moduly, každý z nich má různé možnosti. Je zde také možnost řetězit moduly pro zvýšení efektivnosti. Zde je seznam modulů, které jsou pro VLC dostupné: [9]

- **Standard** umožňuje poslat stream přes přístupový výstupní modul, například UDP, soubor, HTTP, atd. Tento modul se používá na konci řetězce.
- **Transcode** se používá pro transkódování, kódování a dekódování streamu za použití různých kodeků a bitových rychlostí audio nebo video vstupu. Pokud vstupní nebo výstupní přístup nedovoluje kontrolu tempa (síť, kamera), je transkódování realizováno za běhu v reálném čase, to může vést ke zvýšené spotřebě výkonu CPU v závislosti na použitých parametrech. Jiné streamy, jako jsou třeba soubory nebo disky, jsou kódovány co nejrychlejším způsobem, jaký jim systém dovolí.
- **Duplicate** umožňuje vytvořit druhý řetězec, ve kterém stream řeší nezávislým způsobem na tom prvním.
- **Display** nám dovolí zobrazit vstupní stream stejně, jako by to normálně udělal VLC klient. Pokud je využíván zároveň s Duplicate modulem, umožní nám to monitorovat stream za běhu.

- **RTP** umožní stream přes protokol RTP (jeden UDP port pro každý elementární stream). Tento modul také umožňuje podporu RTSP protokolu.
- **ES** dovoluje vytvořit separátní elementární streamy ze vstupního streamu. Toto může být použito například pro uložení audio a video streamů do různých souborů.

Každý z těchto modulů může také používat tzv. Options (možnosti). Zde je syntaxe, která musí být dodržena.

```
vlc input_stream --sout "#module1{option1=parameter1{parameter-option1},option2=
    parameter2}:module2{option1=...,option2=...}:..."
```

Některé možnosti modulu (v našem případě je to option1) musí být nastaveny, jiné jsou dobrovolné (například parameter-option1). Tyto možnosti jsou mnohdy součástí hodně pokročilého nastavení.
Další možná syntaxe pro zápis modulů vypadá takto.

```
vlc input_stream --sout-module1-option1=... --sout-module1-option2=... --sout-module2-
    option1=... --sout-module2-option2=... ...
```

Pro příklad, pro transkódování streamu a odeslání bychom použili tuto sekvenci.

```
vlc input_stream --sout '#transcode{options}:standard{options}'
```

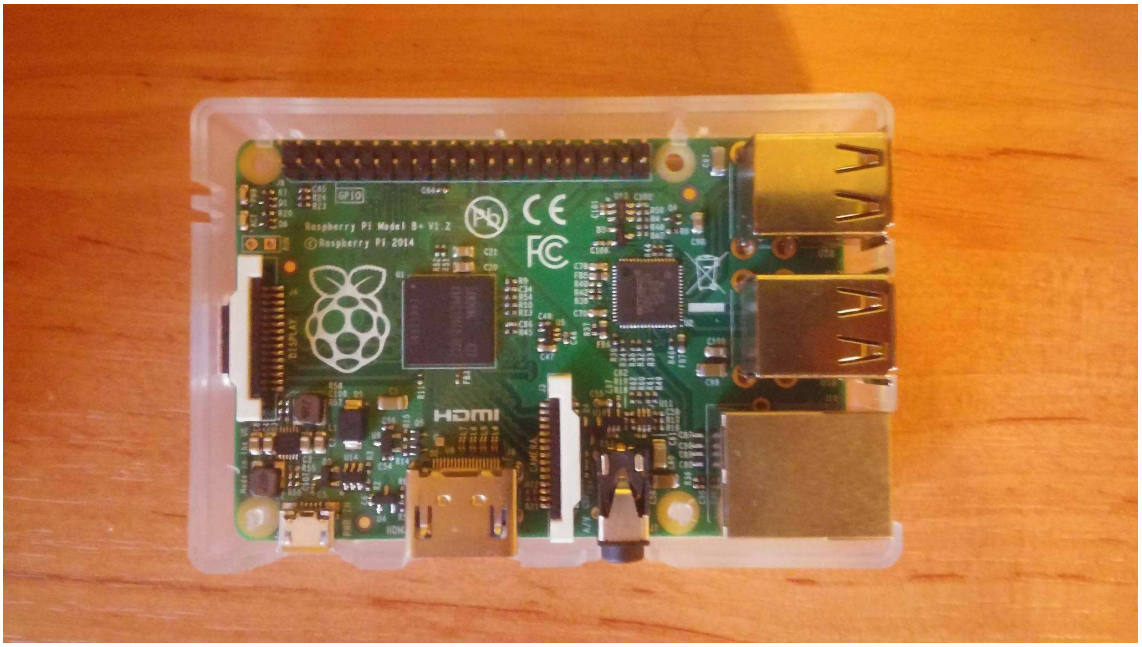
7 Nízkorozpočtový server v prostředí Linux

7.1 Cíl

Hlavním cílem této práce bylo najít vhodné řešení pro cenově nenáročný stream server. Většina měření a konfigurace proběhla v laboratoři EB215, pro testování funkčnosti příkazů byly použity výkonnější PC v laboratoři a posléze byly testovány na Raspberry Pi Model B+ 811-1284, které jsem si vybral pro svůj server.
Bylo také nutno zvolit vhodný program pro stream server, vybral jsem si program VLC, který podporuje mnoho různých kodeků, ale hlavně funguje jako server i klient a podporuje IPv6 adresaci.

7.2 Raspberry Pi

Raspberry Pi mě s ohledem na svou cenu zaujalo jako první. Na českém trhu se jeho cena pohybuje zhruba kolem 899 Kč, je skladné a hlavní je, že na něm funguje systém Raspbian na bázi Linuxu Debian. Raspberry Pi je série levných a malých jednodeskových



Obrázek 7.1: Raspberry Pi Model B+ 811-1284.

počítačů vyvíjen ve Velké Británii společností Raspberry Pi Foundation za účelem učení základních principů počítačů v rozvojových zemích.

Základní parametry Raspberry Pi Modelu B+ 811-1284 jsou:

- Rozměr: 85x21x56mm
- Váha: 0,07 kg
- Procesor: Broadcom SoC běžící na frekvenci 700MHz
- Paměť RAM: 512MB
- 4 USB porty
- 40 GPIO pinů
- 3.5mm jack na sluchátka
- Slot pro Micro SD kartu
- Napájecí konektor microUSB
- HDMI port pro připojení video výstupu

Na Raspberry Pi jsem nainstaloval systém na bázi Linuxového operačního systému Debian, upraven pro Raspberry s názvem Raspbian. Celý systém je uložen na microSD kartě, pro kterou je ve spodní části zařízení slot. Současná verze se jmenuje Raspbian Jessie, kernel verze 4.4 a byla vydána 3.února roku 2017. [16]

7.3 Konfigurace

Kromě programu VLC, který byl potřeba doinstalovat pro systém Raspbian, jsem musel nakonfigurovat systém tak, aby vůbec splňoval podmínky podpory IPv6, což zahrnovalo přidání IPv6 do souboru s moduly, aby se načetl při boot sekvenci. Také bylo nutno upravit nastavení DHCPv6 a DNS a dále nainstalovat programy pro měření serverových parametrů. Pro měření CPU a RAM bylo použito standartního příkazu top -p ”číslo procesu”kde uvedeme PID VLC a pro měření síťových parametrů je použit software Slurm a dále okrajově zachytíme provoz v softwaru Wireshark. [18]

7.4 Stream z mp4 souboru

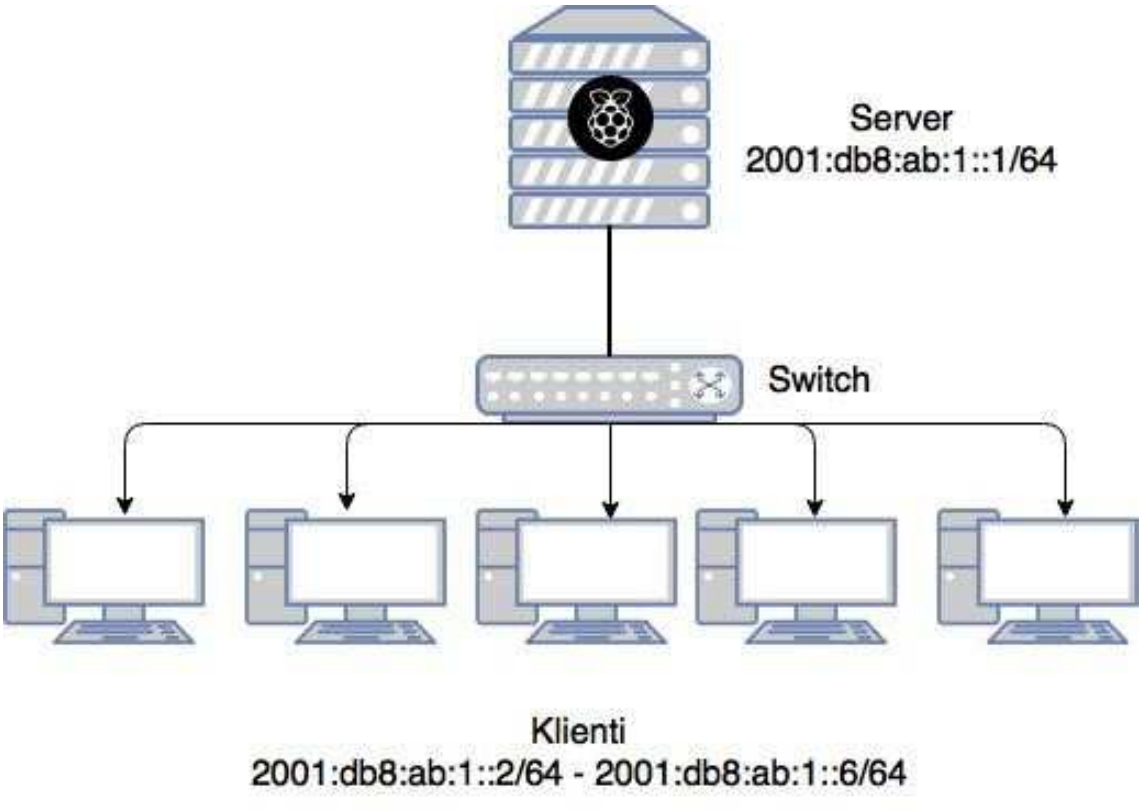
Jako první jsem streamoval soubor s názvem video.mp4, jehož parametry [17] jsou:

- Kodek: H264 - MPEG-4 AVC
- Rozlišení: 1280x720
- Procesor: Broadcom SoC běžící na frekvenci 700MHz
- Snímků za sekundu: 30

Parametry zvukové stopy jsou:

- Kodek: MPEG AAC Audio (mp4a)
- Kanály: Stereo
- Vzorkovací frekvence: 44100 Hz

Schéma zapojení, postupně jsem připojoval 1 až 5 klientů do switche ke kterému byl připojen Raspberry Pi server s IP adresou 2001:db8:ab:1::1/64. Klienti poté obdrželi adresy 2001:db8:ab:1::2/64 až 2001:db8:ab:1::6/64



Obrázek 7.2: Schéma zapojení v laboratoři EB215.

Na klientech i na serveru je vypnut síťový manažer.

```
sudo stop network-manager
```

Adresy se poté přiřadí fixně tímto příkazem.

```
sudo ip addr add 2001:db8:ab:1::1/64 dev eth0
```

Ještě zbývá nastavit rozhraní na up.

```
sudo ip link set dev eth0 up
```

První linuxový příkaz, který byl testován, pouze vyšle data ze souboru na multicast adresu ff15::1 ve formátu v jakém se nachází, není zde provedeno žádné transkódování.

```
vlc −vvv video.mp4 −−sout udp:[ff15::1] −−ttl 12
```

Stream je poté dostupný z klienta přes UDP za použití příkazu.

```
vlc −vvv udp://@[ff15::1]
```

Lze použít i modul Standard, kde specifikujeme zvlášť přístup, mux a destinaci kam mají být data doručena.

```
vlc video.mp4 −−sout '#standard{access=udp,mux=ts,dst=[ff15::1]}' −vvv
```

7.4.1 Hardwarové nároky

Výstup je plynulý, přenese se v pořádku zvuk i video, proces VLC zabírá pouze 7-15% CPU a 7% paměti RAM, po hodinovém vysílání. Po připojování více stanic jsem zjistil, že připojeno může být klidně 50 klientů, spotřeba CPU a RAM bude stále stejná, vysíláme totiž na multicast adresu, odesíláme stejný tok dat, klienti se nepřipojují na server.

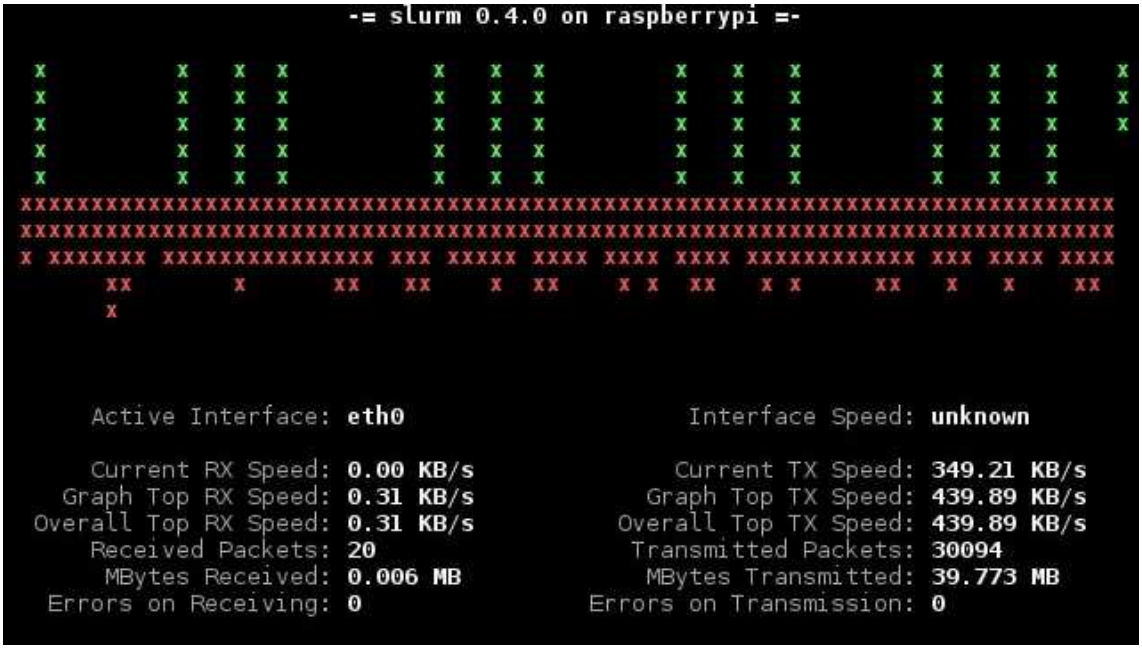
```
top - 07:07:13 up 17 min,  4 users,  load average: 0.46, 0.60, 0.45
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s): 18.6 us,  9.3 sy,  0.0 ni, 65.4 id,  0.0 wa,  0.0 hi,  6.8 si,  0.0 st
KiB Mem:    447996 total,   397340 used,    50656 free,    10940 buffers
KiB Swap:   102396 total,        0 used,   102396 free,   300036 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 2367 pi        20   0 81912  32m  13m S  15.0  7.5   1:30.58 vlc
```

Obrázek 7.3: Provedení příkazu TOP na proces 2367 VLC pro zjištění spotřeby CPU a paměti RAM.

7.4.2 Síťové nároky

Posléze jsem měřil i síťové parametry, za tímto účelem jsem použil program Slurm a měřil jsem jakou rychlostí, kolik paketů a s jakou velikostí proteče za jednotku času 2 minuty. Ve výpisu můžeme vidět i červeně zaznačené odeslané (upload) pakety a zeleně zaznačené přijaté (download) pakety. Při připojení více stanic jsem naměřil stále stejnou hodnotu, odesíláme jen jedny data na konkrétní multicast adresu a ne na jednotlivé klienty. U souboru mp4 se pro přenos audia a videa při nespecifikovaném transkódování použije protokol MPEG TS, který je již popsán v této kapitole 3.6.

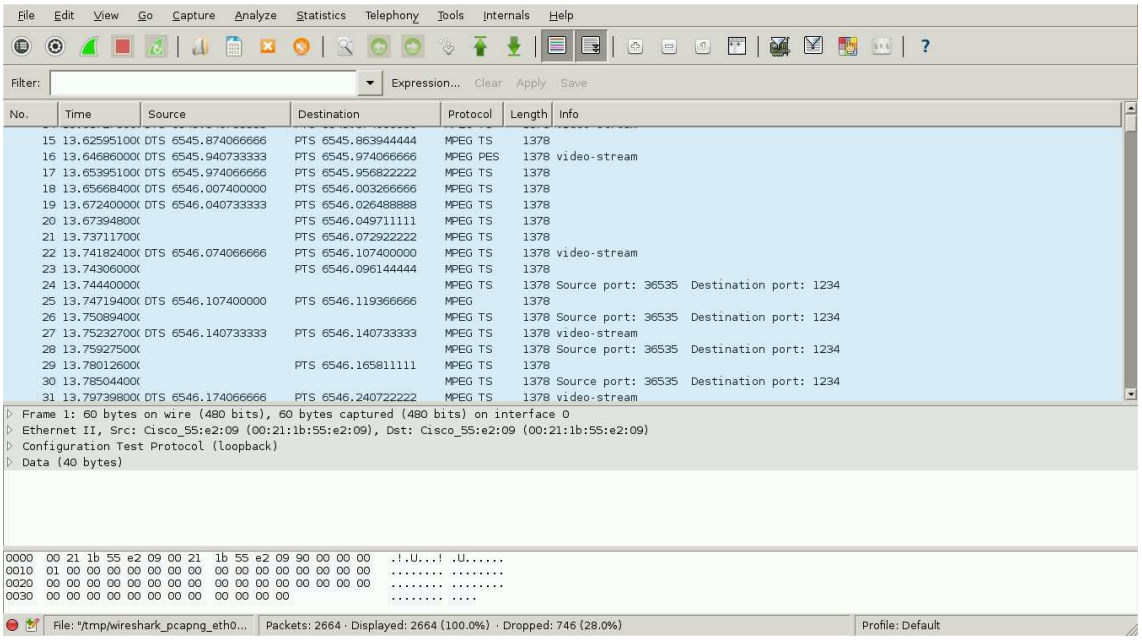


Obrázek 7.4: Provedení příkazu Slurm na Ethernetové rozhraní se spuštěným streamem.

Dále v programu Wireshark můžeme vidět přenos jednotlivých paketů ze serveru 2001:db8:ab:1::1/64 na multicast adresu ff15::1, jako přenosový protokol je použit MPEG Transmision Protocol, který je rozebrán zde 3.6. Paketů se za dobu 2 minut přenese zhruba 30 tisíc. Výpis obsahuje i defaultní porty na kterých komunikuje, kdybychom chtěli, můžeme také port specifikovat v příkazu hned za IP adresou, tímto způsobem.

```
vlc video.mp4 --sout '#standard{access=udp,mux=ts,dst=[ff15::1]:1234}' -vvv
```

Kde :1234 je port na který data odesíláme.



Obrázek 7.5: Zachycení provozu streamu v programu Wireshark.

7.5 Stream z mp4 souboru s transkódováním

Za předpokladu, že bychom chtěli námi vybraný soubor ještě transkódovat do jiného formátu, můžeme použít modul transcode. Mnou testovaný příkaz vypadal takto:

```
vlc −vvv video.mp4 −−loop −−sout '#transcode{vcodec=h264,acodec=mp3,vb=2,ab=2}:std{
    access=udp,mux=ts,dst=[ff15::1]}'
```

Jako transkódovací kodek byl vybrán H264, audio kodek mp3, parametr vb znamená bitrate transkódovaného video streamu v kbit/s, ab poté bitrate transkódovaného audio streamu v kbit/s. Bohužel pokud chceme po Raspberry Pi transkódovat ze souboru, využití CPU VLC se dostane na 50-90%. Celkové využití CPU systému se dostává na 100% a jediné co se přenese je zvuk, který ovšem kolabuje, to i poté co jsem snížil bitrate videa i audia jen na 2 kbit/s. Příkaz byl testován na PC stanici v EB215 s Ubuntu, kvalita videa a audia je s touto formulací velice nízká, nicméně větší výpočetní výkon umožní transkódování. Vyzkoušel jsem ještě jeden příkaz, ve kterém dále specifikuji, že se pro přenos streamu použije protokol RTP. [8]

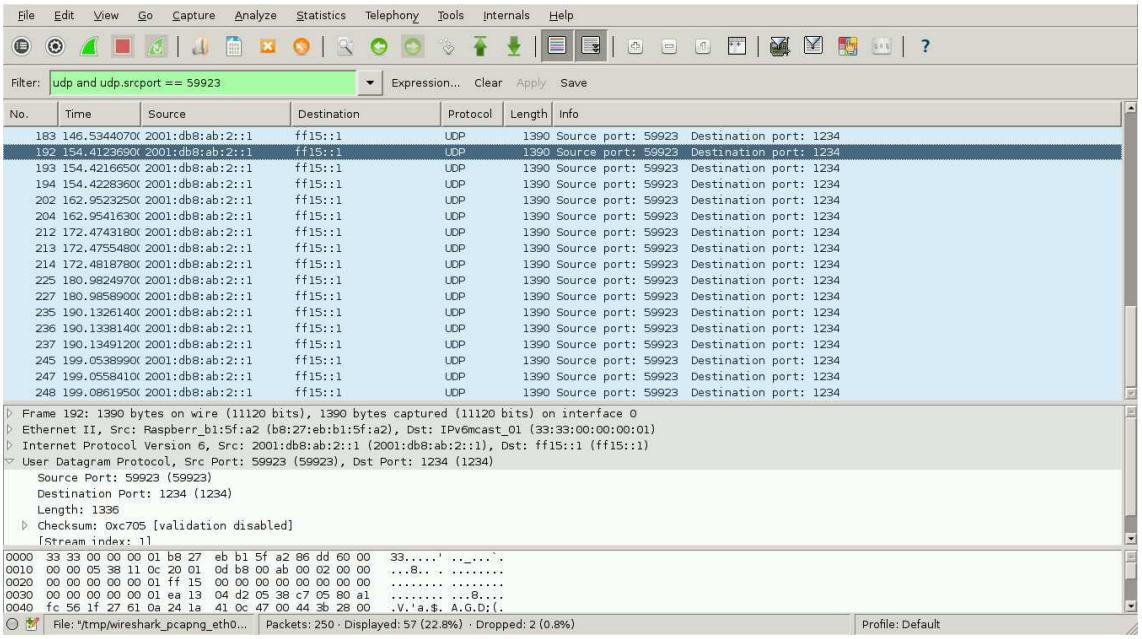
```
vlc −vvv video.mpeg :norm=ntsc :v4l2−width=320 :v4l2−height=240 :v4l2−standard=45056 :
    channel=1 −−no−sout−audio −−sout '#transcode{vb="1600",vcodec=mpgv,acodec=mpga,
    venc=ffmpeg}:rtp{proto=udp,mux=ts,dst=239.255.0.1,port=9001}' −−loop −−ttl 1
```

```
top - 08:14:30 up  1:24,  5 users,  load average: 2.12, 1.80, 1.61
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s): 93.4 us,  6.3 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
KiB Mem:    447996 total,   427756 used,    20240 free,     3556 buffers
KiB Swap:   102396 total,        4 used,   102392 free,   270316 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 2826 pi        20   0  100m  54m 7952 S  50.1 12.4   0:48.42 vlc
```

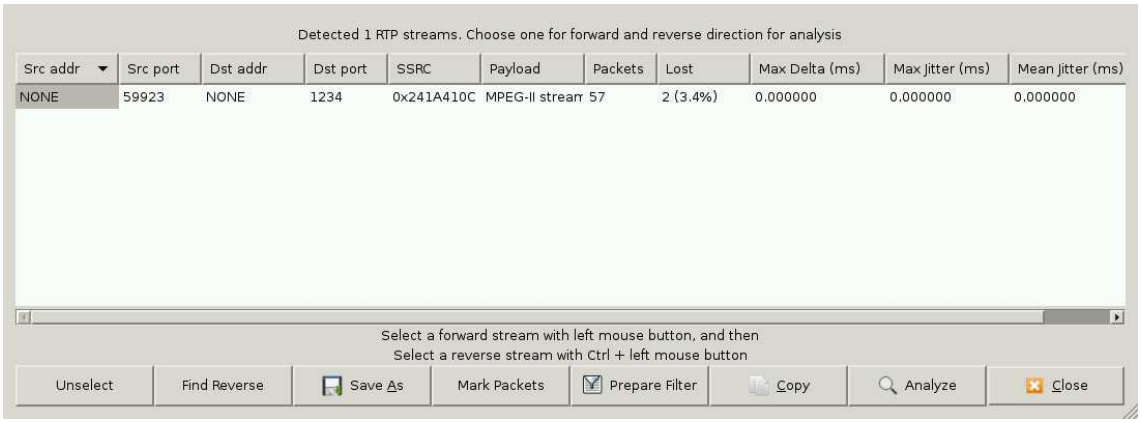
Obrázek 7.6: Provedení příkazu TOP na proces 2826 VLC pro zjištění spotřeby CPU a paměti RAM při transkódování a přenosu RTP.

Příkaz opět běží z Linuxové stanice v laboratoři EB215, ovšem Raspberry Pi nemá výpočetní výkon na to, aby ho zvládlo transkódovat. Stream šel ovšem zachytit v programu Wireshark jako RTP stream. Za celé 2 minuty se přeneslo pouze zhruba 150 paketů. Přenos můžeme vidět zde:



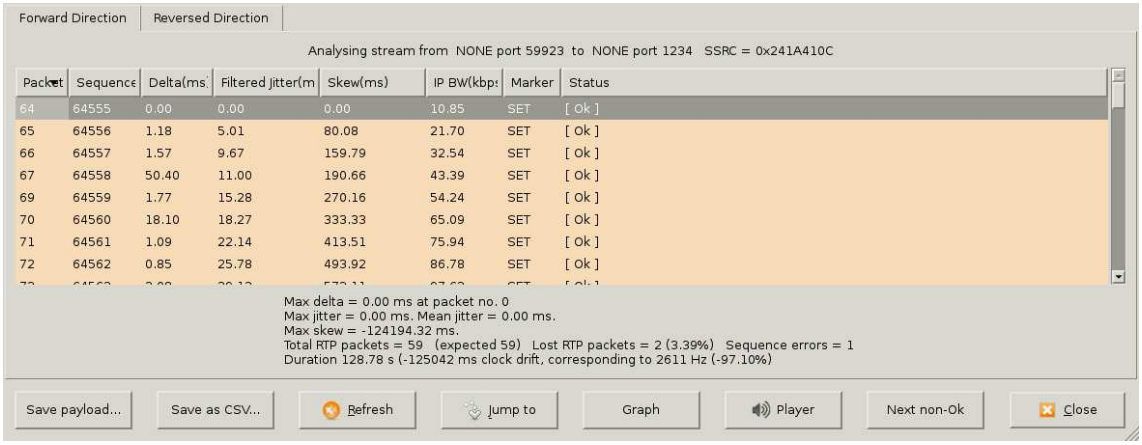
Obrázek 7.7: Zachycení provozu RTP streamu v programu Wireshark.

Následný stream lze poté odchytit zde:



Obrázek 7.8: Detekce RTP streamu v programu Wireshark.

Jeho analýzu můžeme vidět zde:



Obrázek 7.9: Analýza paketů RTP streamu v programu Wireshark.

7.6 Stream Video on Demand

Server můžeme také nakonfigurovat tak, aby nám nestreamoval real time video, ale aby bylo video kdykoliv dostupné tzv. Video on Demand. Server stačí nakonfigurovat pomocí telnetu. Nejprve specifikujeme telnetové RTSP porty a heslo.

```
vlc --ttl 12 -vvv --color -I telnet --telnet-password videolan --rtsp-host 0.0.0.0 --rtsp-
    port 8554
```

Poté se připojíme k telnetu na portu 4212.

```
telnet localhost 4212
```

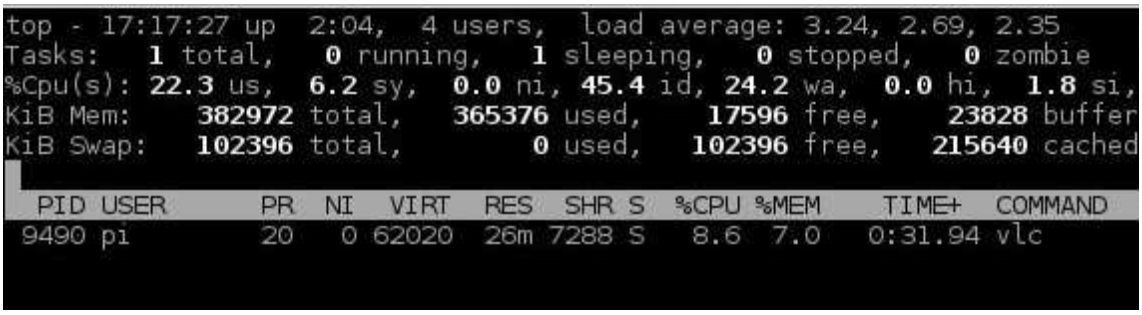
Vytvoříme zde nový objekt.

```
new Test vod enabled
setup Test input video.mp4
```

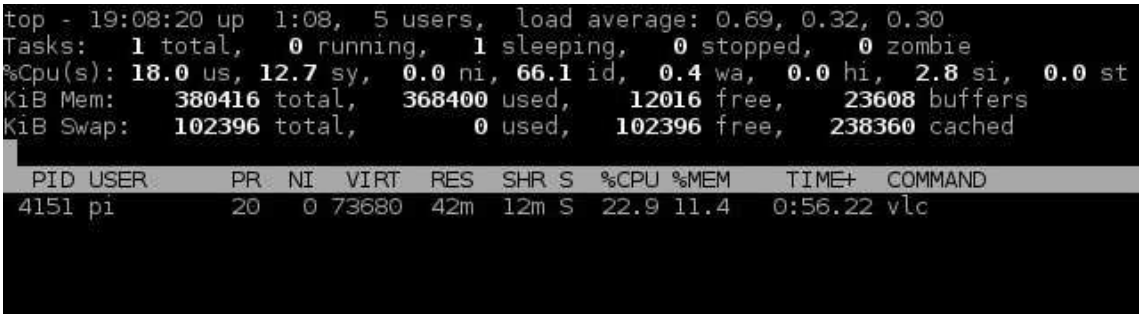
K videu se poté připojíme na klientovi.

```
vlc rtsp://192.168.1.100:8554/Test
```

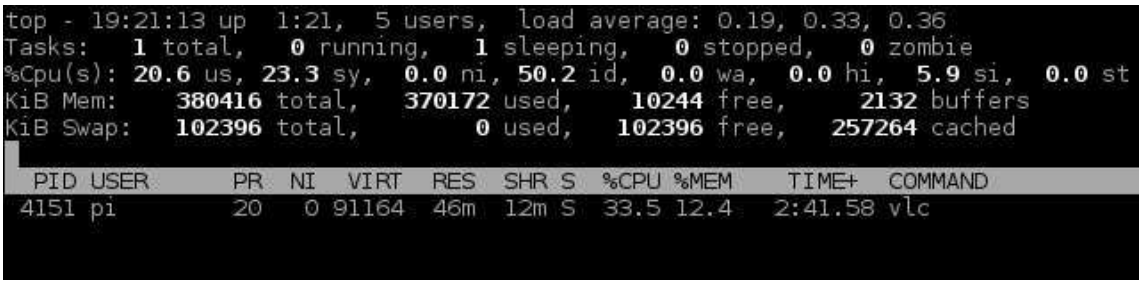
Hostovská IP adresa a IP adresa serveru je ve formátu IPv4, je to z toho důvodu, že pro přenos VoD protokol RTSP nemá a nikdy mít nebude podporu IPv6, protože jeho implementace nedovoluje jak specifikoval tým live555, tento poznatek byl ověřen z více zdrojů. Při Video on Demand také vzrůstá nárok na výkon serveru, každý klient si může vyžádat jiné video, nebo stejné video v jiném čase, server tedy musí posílat data separátně na každou stanicí. Zde můžete vidět, jak se lišila spotřeba zdrojů Raspberry Pi u jednoho, dvou a tří počítačů. [20]



Obrázek 7.10: Provedení příkazu TOP na proces 9490 VLC pro zjištění spotřeby CPU a paměti RAM při konfiguraci Video on Demand s jedním příjemcem.

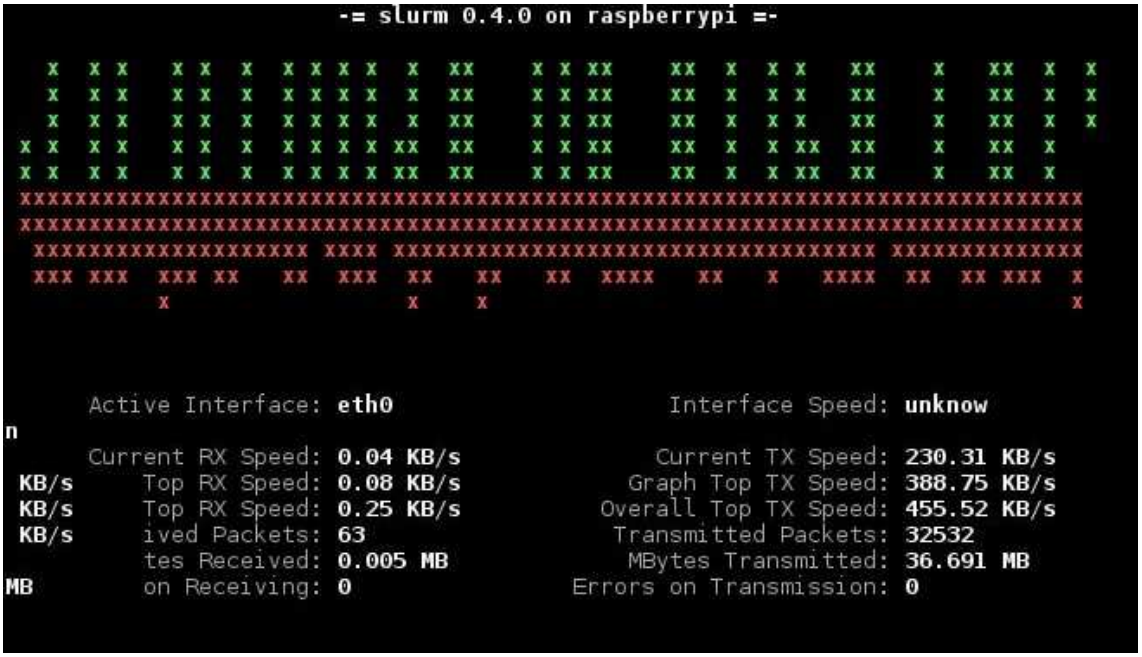


Obrázek 7.11: Provedení příkazu TOP na proces 4151 VLC pro zjištění spotřeby CPU a paměti RAM při konfiguraci Video on Demand se dvěma příjemci.



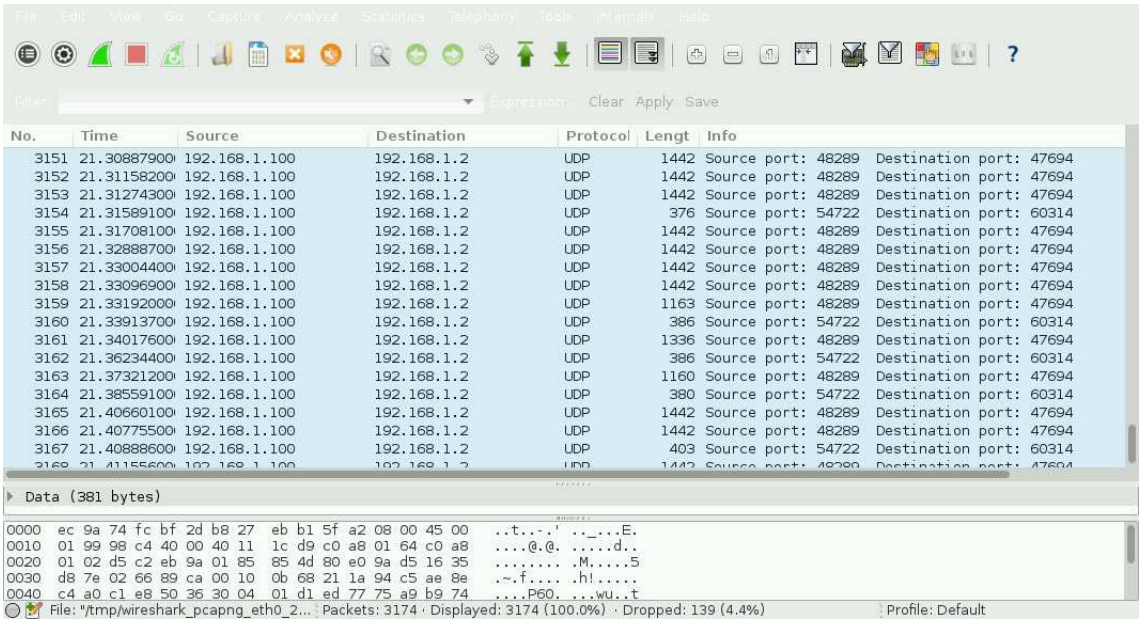
Obrázek 7.12: Provedení příkazu TOP na proces 4151 VLC pro zjištění spotřeby CPU a paměti RAM při konfiguraci Video on Demand se třemi příjemci.

Při síťovém měření můžeme vidět že příchozích paketů je více než u lineárního přenosu z důvodu synchronizace.



Obrázek 7.13: Provedení příkazu Slurm na Ethernetové rozhraní se spuštěným Video on demand.

Stream byl také zachycen v programu Wireshark.



Obrázek 7.14: Analýza paketů VoD streamu v programu Wireshark.

7.7 Stream z web kamery

Máme také možnost streamování live videa z web kamery, kterou připojíme přes USB port, za účelem této práce jsem pracoval s externí web kamerou Logitech C920. Narazil jsem ovšem na problém, který jsem nebyl schopen vyřešit. Web kamera byla detekována v USB zařízeních, ovladače jsem doinstaloval. S normálními příkazy pro pořízení fotografie jako je:

```
fswebcam -r 1280x720 obrazek.jpg
```

Web kamera obrázek pořídí a uloží jej do svého domovského adresáře. Stejně jako když použijeme příkaz pro zachycení video do VLC.

```
vlc v4l2:///dev/video1 --demux h264
```

Web kamera video zachytí na několik vteřin a poté se zasekne z důvodu slabého výpočetního výkonu, po zadání příkazu top pracuje procesor celou dobu na 100%. Což by nebyl takový problém, video nechceme vykreslit na Raspberry Pi, chceme jej pouze nahrát a odeslat.

```
vlc v4l2:///dev/video0 --sout '#standard{access=udp,mux=ts,dst=[ff15::1]}' -vvv
```

Obrázek 7.15: Připojená Web kamera Logitech C920.

Při zadání tohoto příkazu se spustí VLC, bohužel neodesílá žádné pakety, protože není schopen dohledat Stream Input. Alternativou je tedy použít tento příkaz, který nám video nahraje přes web kameru do souboru a soubor posléze odešleme výše zmíněným příkazem pro vysílání ze souboru a nebo pro stream použít program s názvem Motion, který nám umožní live streamovat skrz internetový prohlížeč a HTTP protokol.

```
sudo service motion start
```

Možnosti výstupu můžeme měnit v souboru /etc/motion/motion.conf musíme zde také změnit option z ”Daemon OFF”na ”Daemon ON”a přidat ”ipv6_enable ON”abychom zajistili že stream bude dostupný z IPv6 adresy. Do webového prohlížeče pak již stačí jen zadat IP adresu ve stylu http://[2001:db8:ab:1::1]:8081.

Obrázek 7.16: Ukázka výstupu na straně klienta.

8 Závěr

Cílem mé bakalářské práce bylo provést návrh a testování serveru určeného pro streamování zábavního obsahu v prostředí Linuxu s podporou IPv6. Na začátku bylo provedeno seznámení se současnou problematikou streamování, úvod do IPv6, vysvětlení základních protokolů pro přenos a základních pojmů jako je stream, webcasting, multicasting atd.
Dále byl proveden výběr vhodného hardwarového a softwarového vybavení. Možnosti programu VLC byly otestovány na výkonnější počítačové stanici pro verifikaci validity testovaných příkazů. Dále byl nakonfigurován server na Raspberry Pi a bylo provedeno síťové a hardwarové měření pro streamování lineárně ze souboru, Video on Demand a stream z webkamery. Veškeré poznatky byly uvedeny v této práci.
Zjištěno bylo také, že pro přenos VoD je nevhodný protokol RTSP z důvodů nepodporované IPv6. Raspberry Pi B+ 811-1284 se také potýkalo s problémy co se týče hardwaru u řetězených příkazů, kde bylo například použito transkódování. Pro jednoduchý stream server je tedy Raspberry Pi Model B+ 811-1284 dostačujícím řešením.
Pokud bychom chtěli pokročilejší řešení, tak je vhodné uvažovat nad výkonnějšími hardwarovými zařízeními. Doporučoval bych testování na Raspberry Pi 3 Model B, které má již čtyřjádrový procesor a tyto úkony by mohlo zvládat lépe.
Do budoucna plánuji v práci pokračovat a to s výkonnějším vybavením, a také bych chtěl zkonstruovat webové rozhraní pro Video on Demand server, kde si klienti mohou pomocí grafického rozhraní individuálně vybrat, co si přejí sledovat. Inspiraci pro tento nápad mi poskytl kanál Youtube.com.

Jan Pejsar

9 Seznam použité literatury

Literatura

[1] Linear and Non-Linear Video and TV Applications: Using IPv6 and IPv6 Multicast. 1. USA: John Wiley, 2012. ISBN 1118327462.

[2] Wiki.videolan.org. (2017). Documentation:Streaming HowTo/Command Line Examples - VideoLAN Wiki. [online] Dostupné z: https://wiki.videolan.org/Documentation:Streaming-HowTo/Command-Line-Examples/ [Accessed 8 Dub. 2017].

[3] En.wikipedia.org. (2017). IPv6. [online] Dostupné z: https://en.wikipedia.org/wiki/IPv6 [Accessed 14 Bře. 2017].

[4] Ipv6.org.nz. (2017). New Zealand IPv6 Task Force — FAQs. [online] Dostupné z: http://www.ipv6.org.nz/ipv6-faqs/ [Accessed 19 Bře. 2017].

[5] Peterka, J. (2017). Jak funguje IPTV? - Lupa.cz. [online] Lupa.cz. Dostupné z: http://www.lupa.cz/clanky/jak-funguje-iptv/ [Accessed 14 Bře. 2017].

[6] Videolan.org. (2017). VLC - Features - VideoLAN. [online] Dostupné z: https://www.videolan.org/vlc/features.php [Accessed 15 Bře. 2017].

[7] En.wikipedia.org. (2017). VLC media player. [online] Dostupné z: https://en.wikipedia.org/wiki/VLC-media-player [Accessed 15 Bře. 2017].

[8] Wiki.videolan.org. (2017). Transcode - VideoLAN Wiki. [online] Dostupné z: https://wiki.videolan.org/Transcode/ [Accessed 8 Dub. 2017].

[9] Wiki.videolan.org. (2017). Documentation:Streaming HowTo/Command Line Examples - VideoLAN Wiki. [online] Dostupné z: https://wiki.videolan.org/Documentation:Streaming-HowTo/Command-Line-Examples/ [Accessed 8 Dub. 2017].

[10] Webmarketcentral.com. (2017). Overview of Webcasting and Podcasting. [online] Dostupné z: http://www.webmarketcentral.com/webcast-podcast-overview.htm [Accessed 19 Bře. 2017].

[11] Google.com. (2017). IPv6 – Google. [online] Dostupné z: https://www.google.com/intl/en/ipv6/statistics.htmltab=per-country-ipv6-adoptiontab=per-country-ipv6-adoption [Accessed 19 Bře. 2017].

[12] Tools.ietf.org. (2017). RFC 2326 - Real Time Streaming Protocol (RTSP). [online] Dostupné z: https://tools.ietf.org/html/rfc2326 [Accessed 19 Mar. 2017].

[13] Wiki.videolan.org. (2017). Documentation:Streaming HowTo/Advanced Streaming Using the Command Line - VideoLAN Wiki. [online] Dostupné z:

https://wiki.videolan.org/Documentation:Streaming-HowTo/Advanced-Streaming-Using-the-Command-Line/ [Accessed 19 Bře. 2017].

[14] Afterdawn.com. (2017). MPEG-2 Transport Stream - AfterDawn: Glossary of technology terms a acronyms. [online] Dostupné z: http://www.afterdawn.com/glossary/term.cfm/mpeg2-transport-stream [Accessed 7 Dub. 2017].

[15] En.wikipedia.org. (2017). MPEG transport stream. [online] Dostupné z: https://en.wikipedia.org/wiki/MPEG-transport-stream [Accessed 7 Dub. 2017].

[16] Cz.rs-online.com. (2017). Raspberry PI B+ — Raspberry Pi Model B+ — Raspberry Pi. [online] Dostupné z: http://cz.rs-online.com/web/p/vyvojove-sady-pro-procesory-a-mikrokontrolery/8111284/ [Accessed 7 Dub. 2017].

[17] Centrix-intl.com. (2017). Raspberry Pi Model B+ (B PLUS) 512MB Computer - In Stock!. [online] Dostupné z: http://www.centrix-intl.com/details.asp?Parent2ID=3aproductid=13147 [Accessed 7 Dub. 2017].

[18] Raspberry Pi. (2017). Download Raspbian for Raspberry Pi. [online] Dostupné z: https://www.raspberrypi.org/downloads/raspbian/ [Accessed 7 Apr. 2017].

[19] Tools.ietf.org. (2017). RFC 6951 - UDP Encapsulation of Stream Control Transmission Protocol (SCTP) Packets for End-Host to End-Host Communication. [online] Dostupné z: https://tools.ietf.org/html/rfc6951 [Accessed 8 Dub. 2017].

[20] Instructables.com. (2017). Configure the Software. [online] Dostupné z: http://www.instructables.com/id/Raspberry-Pi-remote-webcam/step2/Configure-the-software/ [Accessed 9 Dub. 2017].

[21] Well beyond streaming video: IPv6 and the next generation television. 5. USA: Netherlands Inc., 2006. ISBN 2226838272.